RESEARCH PLAN PROPOSAL

A Hybrid of Ant Colony Optimization for Join Queries

in Distributed Databases

For registration to the degree of
Doctor of Philosophy

IN THE FACULTY OF SCIENCE

THE IIS UNIVERSITY, JAIPUR

Submitted by

Preeti Sagar Manocha
Enrollment No.: IISU/2010/177

Under the Supervision of

Dr. Swati V. Chande
Principal, Computer Science
International School of Informatics and Management, Jaipur

Department of Computer Science

July 2012

RESEARCH PLAN PROPOSAL

1. TOPIC

A Hybrid of Ant Colony Optimization for Join Queries in Distributed Databases

2. INTRODUCTION

Databases have been in use since the early days of electronic computing for gaining speedy retrieval of data at the expense of flexibility. A Relational Database is defined as a set of logically inter related data stored in the form of rows and columns. An organized set of programs are needed to handle the huge databases created by various organizations. A Database Management System is a software package that controls the creation, maintenance, storage, modification, and extraction of information from data records, files and other objects.

With the advancement of Computer Networks and increase in size of databases, the decentralization of databases is in focus where information can be shared and retrieved from multiple locations. The computing environments are now composed of large number of workstations and mainframes interconnected to each other by high bandwidth communication networks. This has lead to the development of Distributed Database over multiple machines where distribution of the database is Transparent to the users and its need began in early eighties [3]. A Distributed Database is a set of databases stored on multiple computers that typically appears to applications as a single database. A Distributed Database Management System consists of a single logical database split into a number of fragments stored over a number of computers connected by communication networks under the control of Local and Global Database Manager. The data can be distributed over multiple workstations through various techniques like Replication, Fragmentation or a combination of both. The Distributed Database offers greater reliability, availability and improved performance. It enhances the sharing of files, security, expandability and local autonomy of the geographically distributed database of an organization by enforcing local policies regarding the use of database. They allow allows physical parallelism in the processing of queries, placing data closer to its area of need and hence increasing performance.

The users can access data from a Distributed Database Environment by the use of Global and Local Query Execution Applications. A query in relational algebra is usually expresses as a combination of various operators like selection, projection, Cartesian product, union, set difference, joins etc. A Query allows the user to describe the desired data and is responsible for performing the physical operations to produce the result set. Consequently, a query in a Distributed Database can be executed to simultaneously access/ retrieve and modify the data at multiple locations on a network. The query is executed with the help of a well defined Query Language called SQL (Structured Query Language commonly known as Sequel). SQL consists of Data Definition, Data Manipulation, and Transaction Control Commands to access the database. These queries can also be used to retrieve data from multiple databases using the various types of join operations in a Distributed Database. A simple Selection-Projection-Join (SPJ) query essentially represents a single SQL SELECT-FROM-WHERE block with no aggregation or subqueries.

An example of Select-Project-Join Query is given below:

```
SELECT *
FROM orders, inventory
WHERE orders.product = inventory.product;
```

These queries are processing internally in the database environment. Query processing refers to the range of activities involved in extracting data from database, translating them into physical level of the file system and executing them for the desired result. Query Processing is divided into four main phases [4]:

a) Query Decomposition
b) Query Optimization
c) Code Generation
d) Query Execution

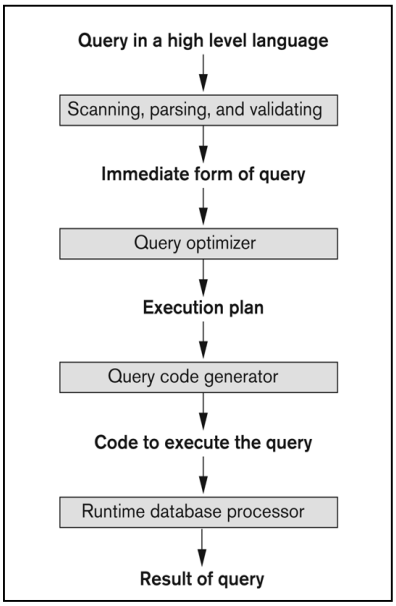


Figure 1. Steps in Query Processing

Query processing in a distributed database requires transfer of data from one computer to another i.e. from remote sites. The time complexity of executing these queries is taken into consideration in designing query optimization algorithms. Query optimization provides speedy, accurate and reliable results to the user with minimum utilization of the system resources. Query optimization is a difficult task in Distributed Environment because of numerous factors like data allocation, speed of communication channel, indexing, availability of memory and many more [19]. It examines all algebraic expressions that are equivalent to the given query and chooses the one that is estimated to be the cheapest. Query Optimization is an important aspect in the execution of queries in a cost effective and efficient method.

2.1 Query Optimization in Distributed Database Management System:

The query optimization problem in large-scale distributed databases is NP-hard in nature and difficult to solve. Query Optimization in Distributed Database allows the system to process more queries in the same amount of time, because each request takes less time than unoptimized queries. It helps the system to reduce the load on the hardware and helps in efficient performance of server by reducing memory usage and lowering power consumption. Query Optimization in Distributed Database consists of four phases [9]:

a) Query Decomposition
b) Data Localization
c) Global Optimization
d) Local Optimization.

Query Decomposition refers to breaking down of a complex query into relational algebra format after query parser, query checks and validation. Data Localization refers to the availability of query data at the local site for processing. The Global Optimization mainly consists of determining the best execution site for local sub query and finding the best inter-site operator scheduling. The Local Optimization is mainly focused on optimizing each local sub query on each site. A lot of research was also made on various factors like optimization algorithms, search space, execution strategies and cost model to optimize a query [1][2][7][8].

Once the query entered by the user is transformed into a standard relational algebra form, the optimizer searches for an optimal query execution plan. Search Space is defined as the set of

all Query Execution Plans that compute the same result. Every point of search space has cost and query execution time associated with it. The query optimizer selects the QEP with least estimated execution cost and time. To achieve this various search strategies are implemented. Search Strategy refers to the enumeration algorithm for the optimizer to determine the best plan to execute the query.

2.2 Join Query Optimization in Distributed Database Environment

Join Operation is one of the most important operations in Distributed Database Environment so as to retrieve data from multiple sites. Join Operation can be considered as a Cartesian product followed by the selection. With the increase in the size of the database and the number of joins, the complexity of the Query Execution Plan and Join Strategy also increases. The complexity of query optimization is determined by a number of alternative QEPs which grows exponentially with the number of relations involved in the query because a single query can be joined in several ways [6]. Since all execution plans are equivalent in terms of their final output with a difference in cost and amount of time that they need to run, it is essential to optimize these query plans, join orders and join methods in modeling query processing. Enumerative optimization strategies are primarily dealing with the join queries. The Join Optimizer plays an important role in query optimization. The task of Query Optimizer is to:

a) Determine the Order of the Execution of Relational Operators.
b) Determine the access methods for pertinent relations.
c) Determine the relations for Join Operations from the given search space by implementing the appropriate search strategy such that the performance measure of the resulting Query Execution Plan is optimized.
d) Determine the order of data movements between the sites so as to reduce the amount of data and cost on the communication network.

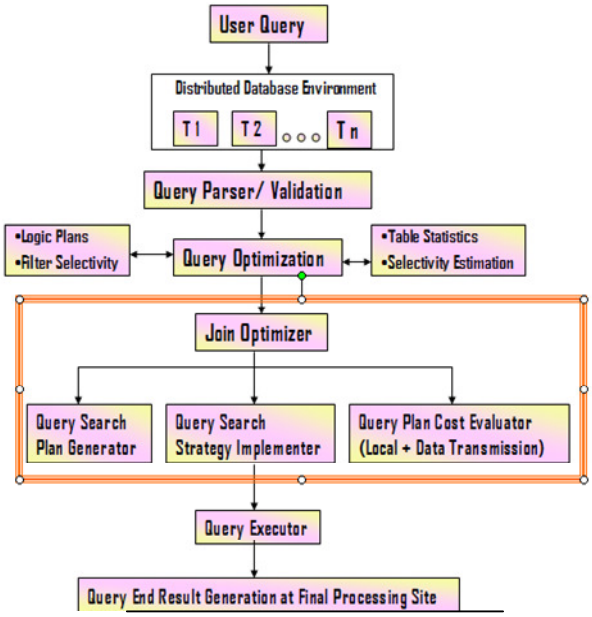


Figure 2. Join Query Optimization in Distributed Database

Research on query optimization algorithms in Distributed Database has been much in focus. A new class of algorithms like Evolutionary Algorithms, Ant Colony Optimization, Particle Swarm Optimization and many more are now been implemented in query optimization of Relational and Distributed Databases to find optimal and sub optimal solutions for large join queries in the given search space because of their global searching capability and their successful application to different combinatorial optimization problems.

2.3 Ant Colony Optimization

Ant Colony Optimization (ACO) first proposed by three Italian scholars, Dorigo M, Colorni A and Maniezzo V in 1992 is a Bionic Optimization Algorithm inspired by Ants that uses probabilistic technique for solving computational problems. As a Distributed Optimization Algorithm, ACO adapts well to serial as well as parallel computers. It is built on the mechanism of positive feedback, so it is very robust, provides intelligent search and can be used for Global Optimization Solutions [13].

ACO is a meta-heuristic, multi agent approach that simulates the foraging behavior of ants for solving difficult NP-hard combinatorial optimization problems like such as Traveling Salesman [15], Quadratic Assignment, Job-Shop Scheduling, Vehicle Routing and many more with

reasonable success. Ants are social insects whose behavior is directed more towards the survival of colony as a whole than that of a single individual of the colony [11]. An important and interesting behavior of an ant colony is its indirect co-operative foraging process. While walking from the food sources to the nest and vice versa, ants deposit a substance, called pheromone trail. Ants can smell pheromone. When choosing their way they tend to choose, with high probability, paths marked by strong pheromone concentration (shorter path) with the result that after some time the whole colony converges toward the use of the path.

Figure 3 given below represents the Flowchart for Ant Colony Optimization Algorithm.

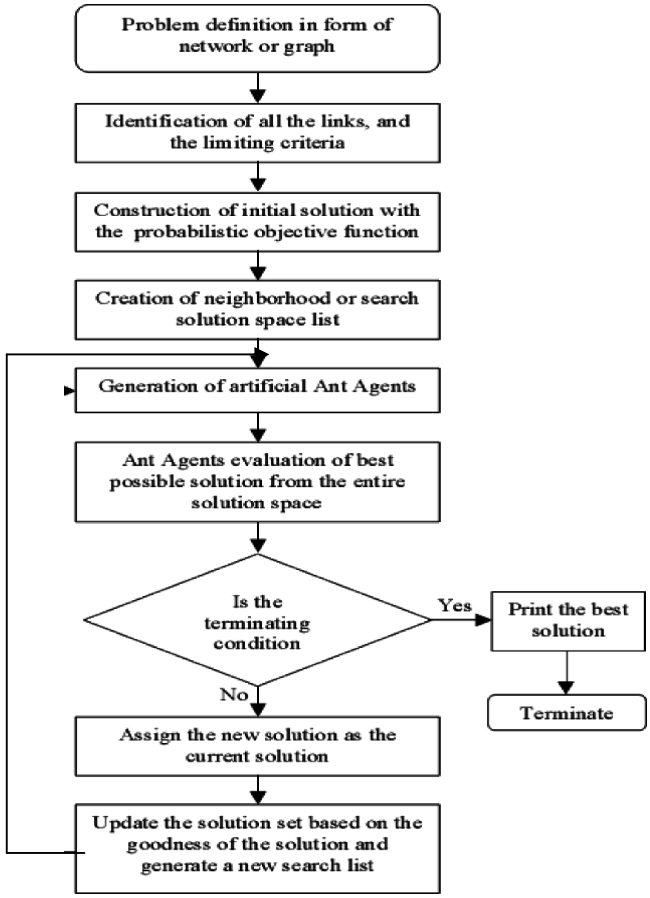
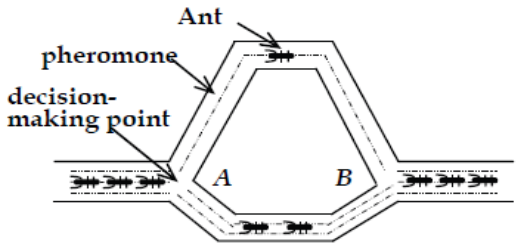

Problem definition in form of network or graph
Identification of all the links, and the limiting criteria
Construction of initial solution with the probabilistic objective function
Creation of neighborhood or search solution space list
Generation of artificial Ant Agents
Ant Agents evaluation of best possible solution from the entire solution space
Is the terminating condition
Yes
Print the best solution
Terminate
No
Assign the new solution as the current solution
Update the solution set based on the goodness of the solution and generate a new search list

Figure 3. Flowchart for Ant Colony Optimization Algorithm

Figure 4 (a,b) given below presents a decision-making process of ants choosing their trips. When ants meet at their decision-making point *A*, some choose one side and some choose other side randomly. Suppose these ants are crawling at the same speed, those choosing short side arrive at decision-making point *B* more quickly than those choosing long side. The ants choosing by

chance the short side are the first to reach the nest. The short side receives, therefore, pheromone earlier than the long one and this fact increases the probability that further ants select it rather than the long one. As a result, the quantity of pheromone is left with higher speed in short side than long side because more ants choose short side than long side. The number of broken line is in direct ratio to the number of ant approximately. Artificial ant colony system is made from the principle of ant colony system for solving kinds of optimization problems. Pheromone is the key of the decision-making of ants [12].



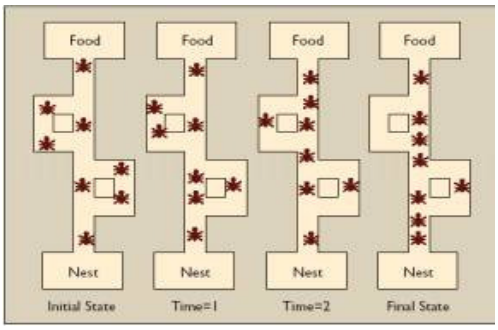


Figure 4(a,b). Ants Traversal Path (a) (b)

2.4 Hybrid of Ant Colony Optimization Algorithm

Inspite of Ant Colony Optimization Algorithm having special characteristics like distributed computing, robust nature and positive feedback mechanism, ACO has some deficiencies:

a) The initial formation needed by ACO has no systematic way of startup.
b) The convergence speed of ACO is lower at the beginning for there is only a little pheromone difference on the path at that time but the convergence speed increases towards optimum answer because of positive feedback mechanism.

Keeping this in mind, various hybrids of ACO with other algorithms like Dynamic Programming, Genetic Algorithm and Particle Swarm Optimization Algorithm have been proposed to provide better results rather than using Ant Colony in isolation.

Tansel et.al. [17] proposed DP-ACO (Dynamic Programming- Ant Colony Optimization) algorithm for the optimization of multi way chain equijoin queries in Distributed Database Environment. Dynamic Programming suffers from long execution times and very large memory requirements as the size of the relations and number of joins increases in the query. Dynamic Programming alone can perform optimization on seven relations but DP-ACO have proved to be viable solution by producing good execution plans with 15 way join queries within limited time

and very limited memory space. Another advantage of DP-ACO is that is that they can be easily adapted to existing query optimizers that commonly use DP-based algorithms.

A new hybrid algorithm of Genetic Algorithm and Ant Colony Optimization was introduced to solve the problems of optimization of join ordering (only nested loop joins considered) in relational database queries [18]. GA has strong adaptability, robustness, quick global searching ability with higher population scattering ability for extensive amplitude of answers. It has such disadvantages as premature convergence and low convergence speed towards optimum answer. On other hand ACO has low population scattering ability, high convergence speed, parallel processing and global searching ability with a positive feedback mechanism.

The author proposed a hybrid of ACO-GA to overcome the shortcomings of both the algorithms. The algorithm adopts Genetic Algorithm to give pheromone to distribute. And then it makes use of ant colony algorithm to give the precision of the solution. This is achieved by first creating a set of execution plans by Artificial Ants. These plans are considered as initial populations of GA. Pheromones in routes are updated. In each iteration, each ant firstly seeks answers, these answers are passed into GA for recombination and the new generation is again passed to the ant colony. This improves the convergence rate of the produced offspring in each generation and also produces an answer closer to the optimum one.

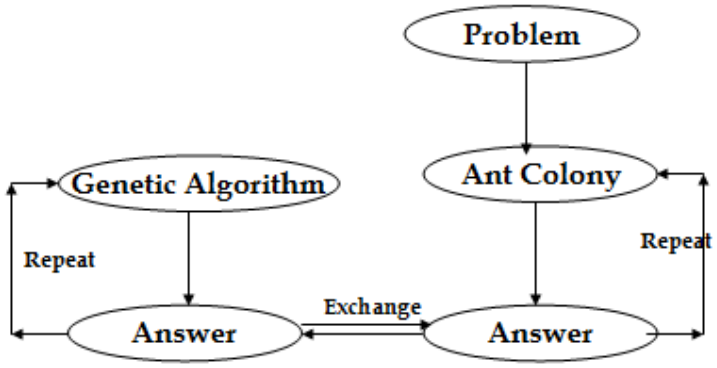


Figure 5. Diagram for Proposed Hybrid GA-ACO Algorithm

By use of the properties of Ant Colony Algorithm and Particle Swarm Optimization, a hybrid algorithm is proposed to solve the traveling salesman problems [20]. The algorithm first adopts statistics method to get several initial better solutions and in accordance with them, gives information pheromone to distribute. Then it makes use of the ant colony algorithm to get several

solutions through information pheromone accumulation and renewal. Finally, by using across and mutation operation of particle swarm optimization, the effective solutions are obtained. The Hybrid Algorithm of ACO-PSO has proved to be effective.

The Search strategy adopted by the Query Optimizer in Distributed Database Management System can help to reduce the query execution time and the cost incurred on the query and hence increases performance of a query by selecting the best Query Execution Plan. The implementations[17][18] of these probabilistic algorithms have proved to generate viable solutions when the size of the query and the number of joins in the query grows.

3. REVIEW OF LITERATURE

All relational database application programs are written in High Level Languages integrating relational language with a declarative interface like SQL to access the data stored in a database. It has been pointed out in [1], that the processing of distributed execution plan from the query via a computer network takes place in three phases:

a) Local Processing Phase or Materialization Phase for all selections and projections
b) Reduction Phase for reducing the size of relations and cost of communications
c) Final Processing Phase or Transmission Phase to generate the output.

An important objective in query optimization in distributed query is to reduce the amount of data transmission required, to reduce the time taken to execute a query and to reduce the cost of query execution for the Reduction and Final Processing Phase [2]. The work related to query optimization in databases started in late seventies which motivated a large part of database scientific community. The optimizer's role is to generate an optimal execution plan from the considered search space so as to minimize response time and maximize throughput with minimum optimization cost. The cost of query execution over multiple locations in a Distributed environment is described as the sum of local cost (Input / Output cost, Cost of processor i.e. CPU cost at each site) and the cost of transferring data between sites. The complexity and cost increases with the increasing number of relations in the query.

Join Query Optimization in Distributed Database implements two important methodologies [5]:

a) Determining an appropriate Join Method for each join operator by taking into account the size of the relations and their access path
b) Generating the order in which the joins are performed with respect to the cost model.

The joining of the relations can take place directly by moving one or more relations to the final site. In addition to joins, semi join operation can also be used as reducers in processing distributed queries [7]. In Distributed Query Processing, the conventional approach to reduce the amount of data transmission is to first apply a sequence of semi joins as "reducers" and then ship the resultant relations to the final site to carry out the join- operations so as to exploit parallelism and minimize processing overhead.

The author reviewed the traditional Static Optimization Strategies like SDD-1, R*, Dynamic Optimization Strategies and Randomized Strategies like Iterative Improvement Algorithms for non-autonomous distributed database systems and analyze the suitability of these strategies for autonomous systems [10]. They further reviewed Mariposa, Query trading with DP and IDP and Query Trading with Processing task Trading (QTPT) for autonomous distributed database systems.

A combination of Best-Worst Ant Algorithm and Genetic Algorithm to achieve high convergence speed and low execution time in Multi-Join Query Optimization in relational database to seek the best join order among the tables [13]. Multi-Join Query Optimization in database is of great necessity to improve the performance of database. Since the query strategy space increases sharply with the growth in the number of connections, it becomes difficult to find optimal solution with minimum time and maximum performance. Hence, the positive feedback mechanism of ACO is combined with the global search capability of GA to generate a hybrid of ACO-GA.

As proposed in [14], the author created a fusion algorithm of Genetic Algorithm and Ant Colony Optimization Algorithm to resolve the optimal solution of bi-level multi-objective optimization model under environmental objective. It utilizes genetic algorithm to initialize pheromone distribution and ant colony algorithm to ensure the precision of the solution. This algorithm is build to overcome the shortcoming of the Ant Colony like lack of initial pheromone and lower convergence speed at an initial phase and the shortcomings of Genetic Algorithm like inability to

utilize the system feedback sufficiently. It makes use of the advantage of the two algorithms to expect the as well. The global searching of genetic algorithm and the positive feedback of ant colony algorithm are working together to make fusion algorithm converge faster to gain optimal solution, optimization precision and optimization speed.

The experimental results of the implementation of GA and ACO to Traveling Salesman Problem were compared by the scientist [15]. The results stated that if the size of the cities are more than 30, genetic algorithms' searching capability will gradually decline to a certain extent, when the number of the cities are too big , it can not obtains the optimal solution in finite iteration , because iterative times are too long and unbearable. Here ant colony algorithm is better than genetic algorithm, and it can reach the optimum value. When the size of cities is too big, ant colony algorithm may appear stagnation. The author suggests a hybrid of GA and ACO to be implemented to achieve optimal results.

In [16], the author frames the join ordering query optimization problem as a Traveling salesman problem for Large Scale Databases and explores several heuristics and Genetic Algorithm to solve the optimization problem large join queries in Large-Scale Databases. The computational results prove that Genetic Algorithms produce viable solutions to the given problem.

4. MOTIVATION/ JUSTIFICATION AND RELEVANCE

Query processing in a Distributed Database requires transfer of data from one computer to another through a communication network. Query at a given site might require data from remote sites. The performance of the distributed database query depends on how fast and efficiently data can be retrieved from multiple sites. Several factors impact the performance of distributed query processing. These factors include:

a) Selection of appropriate site when data is replicated/fragmented at multiple sites.
b) Order of Query Operations like selection, projection, join and transmission.
c) Selection of Join Method / Search Strategies to accelerate the speedy execution of the queries.
d) Selection of Join Order to reduce the number of Search Spaces and to create the best Query Execution Plan.

Research is being extensively carried out for query optimization in distributed databases with large query size and multiple joins. Query optimization involves two main tasks:

a) Search Space generation or generation of Query Execution Plan

b) Finding an optimal plan from the search space, using search strategies and cost model. Numerous optimization strategies like static, dynamic and randomized strategies have been proposed for determining optimized solutions for specific types of queries. Continuous research is being carried out by researchers to reduce the time taken to execute a query and to reduce the cost incurred during the execution of the query in distributed database environment is increasing each day.

When the size and complexity of the query increases, the problem to estimate the optimized solution for the query execution also increases. The research has shown that implementation of an improved Ant Colony Optimization Algorithm have proved to be viable solutions to generate optimal solutions. There is still a lot of opportunity to generate optimized solution for the Join Query in a Distributed Database with a number of parameters influencing the query. Refined Search strategies are needed to generate solutions to such problems. The need of the industry for generating quicker methods for faster solutions for large distributed databases motivates me to create Optimal Join Solutions for query execution in distributed database environment by applying a hybrid of Ant Colony Optimization Algorithm.

5. OBJECTIVES

The main objectives to conduct the research are:

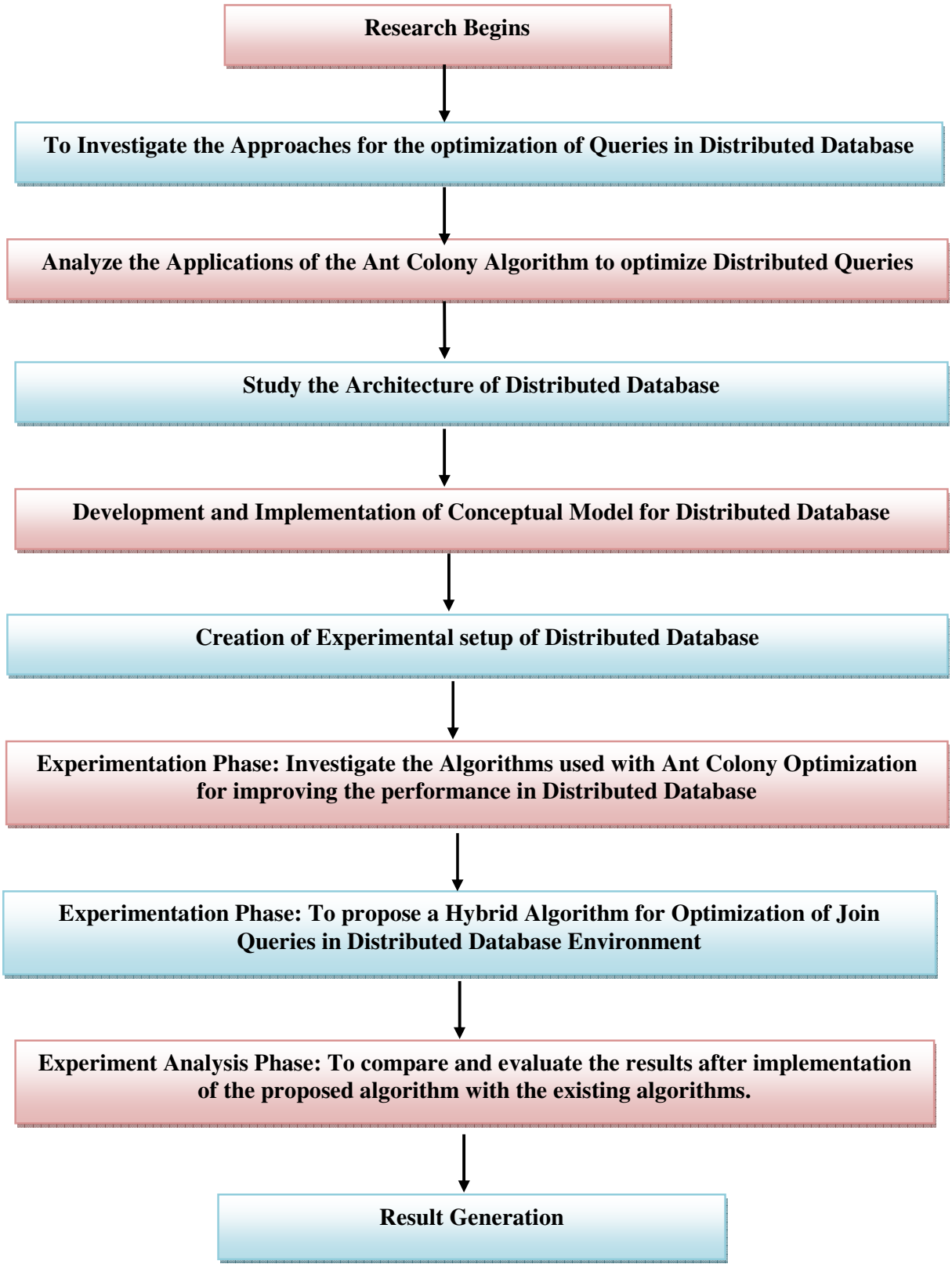
- To investigate the approaches towards the optimization of queries in Distributed Database Environment.
- To study the applications of the Ant Colony Optimization Algorithm to optimization of database queries in Distributed Database Environment.
- To suggest necessary improvements in existing studies of Ant Colony Optimization Algorithm of Join Queries in Distributed Database.
- To investigate the algorithms used with Ant Colony Optimization Algorithm for improving performance of Distributed Database Queries.
- To propose a hybrid algorithm for optimization of join queries in Distributed Database Environment.
- To compare and evaluate the results after implementation of the proposed algorithm with the existing optimization algorithms.

6. PLAN OF WORK AND METHODOLOGY:

a) RESEARCH DESIGN

The query experiment of Distributed Database shall be created interconnected to each other via communication network. The Operating System used for the development shall be Linux / Windows. One of the work stations shall be converted into a Global Work Station. The Front End Development Environment and a Back End Relational Database Management System shall be installed on all the work stations.

b) METHODOLOGY:


Research Begins
To Investigate the Approaches for the optimization of Queries in Distributed Database
Analyze the Applications of the Ant Colony Algorithm to optimize Distributed Queries
Study the Architecture of Distributed Database
Development and Implementation of Conceptual Model for Distributed Database
Creation of Experimental setup of Distributed Database
Experimentation Phase: Investigate the Algorithms used with Ant Colony Optimization for improving the performance in Distributed Database
Experimentation Phase: To propose a Hybrid Algorithm for Optimization of Join Queries in Distributed Database Environment
Experiment Analysis Phase: To compare and evaluate the results after implementation of the proposed algorithm with the existing algorithms.
Result Generation

WORK PLAN:

Phase 1: In depth study of the most recent techniques adopted by the scientist for the optimization of query in Distributed Database Environment shall be done in first 2-3 months.

Phase 2: The applications of Ant Colony Optimization Algorithm for Distributed Database Queries shall be analyzed in the next three months.

Phase 3: Next 2-3 months shall be needed for investigating the algorithm used with ACO for improving the performance in DDBMS.

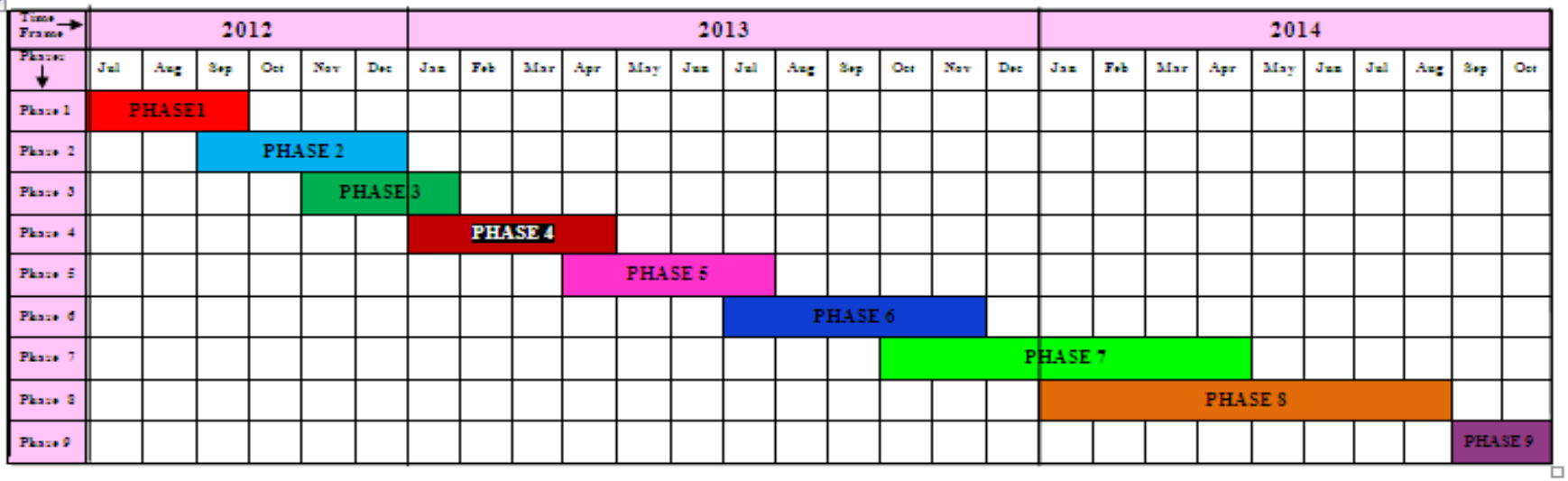
Phase 4: Development of Conceptual Data Model of Distributed Database shall take place in the next 2-3 months where relations, schema and their interconnection shall be defined.

Phase 5-6: Next 6-8 months shall be needed for studying the architecture and creating of experimental setup of DDBMS. A new Algorithm shall be proposed and implemented for the optimization of join queries in Distributed Environment.

Phase 7: Comparative analysis and evaluation of the proposed Hybrid ACO with the existing optimization algorithms along with generation of results shall take place in the next 4-5 months.

Phase 8: Next 4-5 months shall be needed for Thesis writing and its submission.

WORK PLAN



Phase 1: To investigate approaches for the optimization of queries in Distributed Database

Phase 2: Analyse the applications of Ant Colony Optimization Algorithm for Distributed Queries

Phase 3: Investigate the algorithm used with ACO for improving the performance in DDBMS

Phase 4: Development an Implementation of Conceptual Model for Distributed Database Environment

Phase 5: Study the Architecture and creation of experimental setup of Distributed Database Environment

Phase 6: Propose a Hybrid ACO Algorithm for optimization of Join Queries in Distributed Database

Phase 7: Compare and evaluate the results of the proposed algorithm with the existing optimization algorithm

Phase 8: Result Generation and Thesis Writing

Phase 9: Thesis Submission

TOOLS AND TECHNIQUES

1. Time and Cost Estimation Tools:

Different relational databases use different types of tools, functions and commands to estimate the time taken to execute a query and hence to measure the performance of the query. The following tools are used for evaluating the performance of database queries. On the basis of the DDBMS used for experimentation, these tools may be used.

a) db2Batch which works on command line to sort out compile time and execution time of a query is used in DB2 environment.

b) SQL Profiler Tool is used in SQL Server 2008 to measure the performance of query. Also T-SQL scripts using SSMS Query Editor in Test Environments can be used to find out the SQL Execution time in milliseconds.

c) EXPLAIN [ANALYZE] [VERBOSE] statement is used in PostgreSQL to estimate the total elapsed time expended within each plan node in millisecond and also displays the total number of rows actually returned as display.

d) SetQueryTimeout() function or MYSQLCC is applied in mysql to estimate the Query Execution Time.

e) Oracle uses a command line statement: dbms_utility.get_time to estimate the time of the query execution. Also SET TIMING ON is used to estimate the time taken by the query for execution.

Various cost functions are also available with the query optimizer of the relational databases to estimate the cost incurred for the execution of the query. In distributed database environment, the additional cost of transferring the data over the communication channel is also important. The cost involved shall be estimates using the cost function of the given database. For example EXPLAIN PLAN Statement is used in Oracle to estimate the cost of the query execution.

2. Query Standard Tool:

On the basis of the use of database environment in the experimentation, appropriate tools will be used. TPC Benchmark is one of the most commonly used tool for measuring the query standard in a database environment.

- TPC Benchmark **(TPC-H)** is a decision support benchmark**.**

- The performance metric TPC-H is the Composite Query-per-Hour Performance Metric **(QphH@Size)**, and reflects multiple aspects of the capability of the system to process queries.
- It examines large volumes of data, execute queries with a high degree of complexity and give answers to critical business questions.

3. Graphical Tools: MS Excel, Open Office Spreadsheets

7. PLACE OF WORK AND FACILITIES AVAILABLE

The experimental setup for the Distributed Database Environment shall be set up in the Information Technology Lab in The IIS University Campus.

8. REFERENCES, BIBLIOGRAPHY, WEBLIOGRAPHY

[1] C.T. Yu and C.C. Chang, "Distributed Query Processing" ACM Computational Surveys, vol. 16, no.4, pp. 399-433, Dec. 1984

[2] P.M.G. Apers, A.R. Hevner and S.B Yao, "Optimization Algorithm for Distributed Queries", IEEE Trans. On Software Eng., Vol.SE-9, no.1, pp 56-68, Jan. 1983

[3] P.A. Bernstein, N.Goodman, E.Wong, C. Reeve and J.B. Rothnie, " Query Processing in a System for Distributed Database(SDD-1)", ACM Trans. Database Sys, Vol.6, no. 4, pp 602-625, Dec. 1981

[4] A.R. Hevner and S.B.Yao, "Query Processing in Distributed System", IEEE Trans., Software Eng., vol.SE-5, pp 177-187, May 1979

[5] S. Pramanik and D. Vineyard, "Optimizing Join Queries in Distributed Database", IEEE Trans., Software Eng., vol.14, pp 1391-1326, Sept. 1988

[6] M.S.Chen and P.S. Yu, "Using Join Operations as Reducers in Distributed Query Processing", Proceedings of the 2nd International Symp. on Databases in Parallel and Distributed System, July 1990

[7] M.S.Chen and P.S. Yu, "Interleaving a Join and Semi Join Operations in Distributed Query Processing", IEEE Trans. Parallel and Distributed Systems, IEEE Transactions, Vol. 3 No. 5, September 1992, Current Version: 2 Aug 2002

[8] Abdelkader Hameurlain and Frank Morvan, "Evolution of Query Optimization Methods", Trans on Large Scale Data and Knowl. Cent. Sys, I LNCS 5740, pp 211-242, 2009

[9] Ozsu M.T. and Valdureiz P: "Principles of Distributed Database System", 2nd Edition, Prentice Hall, Woodliffs, 1999

[10] Doshi P. and Raisinghani V., "Review of Dynamic Optimization Strategies in Distributed Database", Electronics Computer Technology(ICECT), 3rd International Conference, April 2011

[11] Marco Dorigo, Mauro Birattari, and Thomas Suiitzle, "Ant Colony Optimization- Artificial Ants as a Computational Intelligence Technique", IEEE Computational Intelligence Magazine, November 2006

[12] Enxiu Chen1 and Xiyu Liu, "[Book] Ant Colony Optimization - Methods and Applications-Multi-Colony Ant Algorithm", Google Scholar, April 2011
http://cdn.intechweb.org/pdfs/13584.pdf

[13] Yanfei Zhou, Wanggen Wan, Junwei Liu," Multi-joint query optimization of database based on the integration of best-worst Ant Algorithm and Genetic Algorithm", Wireless Mobile and Computing (CCWMC 2009), IET International Communication Conference, pp 543, December 2009

[14] Zhou Shen-Pei1, Yan Xin-Ping, "The Fusion Algorithm of Genetic and Ant Colony and Its Application", Fifth International Conference on Natural Computation, IEEE Computer Society, 2009

[15] Kangshun Li, Lanlan Kang, Wensheng Zhang, Bing Li, "Comparative Analysis of Genetic Algorithm and Ant Colony Algorithm on Solving Traveling Salesman Problem", IEEE International Workshop on Semantic Computing and Systems, 2008

[16] Zehai Zhou, "Using Heuristics and Genetic Algorithms for Large Scale Database Query Optimization", Journal of Information and Computing Science, Vol 2, No 4, pp- 261-280, 2007

[17] Tansel Dokeroglu, Ahmet Cosar, "Dynamic Programming with Ant Colony Optimization Metaheuristic for optimization of Distributed Database Queries", ISCIS:26th International Symposium on Computer and Information Sciences, IEEE, Vol 2 , pp.107-113, 2011

[18] Kadkhodaei, H. ,Mahmoudi, F.Granular, "A Combination Method for Join Ordering Problem in Relational Databases Using Genetic Algorithm and Ant Colony", IEEE International Conference on Granular Computing, pp 312 – 317, Nov 2011

[19] Craig S. Mullins, "Distributed Query Optimization", Technical Support, July 1996

[20] Gao Shang' 2, Jiang Xin-zil, Tang Kezong', Yang Jingyu, "Hybrid Algorithm Combining Ant Colony Optimization Algorithm with Particle Swarm Optimization", Proceedings of the 25th Chinese Control Conference, Harbin, 7-11 August, 2006